

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-110391

(43)Date of publication of application : 23.04.1999

(51)Int.Cl.

G06F 17/30

G06F 12/00

G06F 17/27

G06F 17/21

(21)Application number : 09-265455

(71)Applicant : HITACHI LTD

(22)Date of filing : 30.09.1997

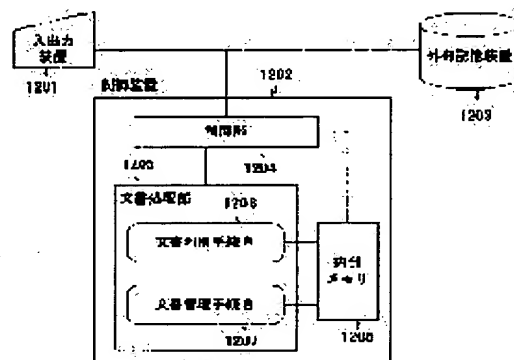
(72)Inventor : SATO YOSHIFUMI
AKISAWA MITSURU
TAKAHASHI TORU
WAKAYAMA SATORU
AZUMA AKIO

(54) DOCUMENT MANAGING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To prevent the number of transfer elements from being increased corresponding to the number of elements by referring to the top node of tree structure from the single transfer element expressing the internal structure of a document.

SOLUTION: A control part 1204 of a controller 1202 controls series of processing related to document processing. Namely, according to a document data processing request inputted from an input/output device, a procedure held in a document processing part 1205 is expanded in an internal memory 1208, document processing is executed, the document data required for the process of document processing are extracted from an external storage device 1203, and processing until the output of the requested processing result is outputted to an input/output device 1201 is controlled. Since the logical structure of the document is expressed as the tree structure of an object group, an application can refer to the tree type logical structure. Besides, the element as the root of logical structure is referred to through the single transfer element expressing the internal structure of the document.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

(11)特許出願公開番号

(43)公開日 平成11年(1999)4月23日

審査請求 未請求 請求項の数 4 O L (全 26 頁)

最終頁に続く

故聖明之君無不任之

【特許請求の範囲】

【請求項 1】 Transfer Element クラスのオブジェクトによって文書の複数の構成要素を管理する DMA (Document Management Alliance) の規定する文書管理方法であって、

論理的な構造を陽に持つ構造化文書に現れる概念を、DMA ベースクラスのサブクラスとして表現し、そのクラスのインスタンス間を関連付けて文書全体の論理的な木構造を構築する手段と、

Transfer Element クラスのオブジェクトが、文書全体の論理的な木構造を表現するノードオブジェクト群の中の、木構造の根となるオブジェクトを参照する手段とを有することを特徴とする文書管理方法。

【請求項 2】 Transfer Element クラスのオブジェクトによって文書の複数の構成要素を管理する DMA (Document Management Alliance) の規定する文書モデルのクラス構成であって、

DMA ベースクラスのサブクラスとして論理的な構造を陽に持つ構造化文書に現れる概念を表現するノードオブジェクトのクラスを具備する文書モデルを用いることを特徴とする文書管理方法。

【請求項 3】 構造化文書に現れる概念を表現するノードオブジェクトのクラスとして、少なくとも論理的な構造の単位となる要素を表現するためのオブジェクトのクラスと、データを表現するためのオブジェクトのクラスとを具備する文書モデルを用いることを特徴とする請求項 2 の文書管理方法。

【請求項 4】 構造化文書に現れる概念を表現するノードオブジェクトのクラスとして、論理的な構造の単位となる要素を表現するためのオブジェクトのクラスと、論理的な構造の単位となる要素に付随する属性を表現するためのオブジェクトのクラスと、外部データの参照を表現するためのオブジェクトのクラスと、テキストデータを表現するためのオブジェクトのクラスと、非テキストデータを表現するためのオブジェクトのクラスとを具備する文書モデルを用いることを特徴とする請求項 2 の文書管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、コンピュータ上における電子化文書の管理システムに係り、特に論理的な構造を明示的な形で表現する構造化文書を含む文書集合を管理する方法に関するものである。

【0002】

【従来の技術】 今後の標準的な文書管理の枠組みを構築するために検討されているアーキテクチャの一つに、DMA (Document Management Alliance) アーキテクチャがある。DMA は、文書処理システムに関連するユーザとベン

ダのほとんどの有力企業が参加しているコンソーシアムであり、文書管理システムの業界仕様の標準化を推進している。DMA アーキテクチャの目指す目的の一つが「文書モデルとそれを操作するためのアプリケーション・インタフェースを統一することによる、オープンなアプリケーションシステム・アーキテクチャの実現」であり、そのための標準的な文書モデルが提案されている。DMA とその文書モデルについては、DMA のホームページ (URL "http://www.aiim.org/dma/") で詳しく解説されている。ここではまず、DMA において規定されている文書モデルについて説明する。

【0003】 図1に、文書オブジェクト及び階層化されたディレクトリ構造を表現するDMAの文書モデル(以下「DMA文書モデル」と呼ぶ)を示す。101は“Container”オブジェクトであり、文書を分類管理するためのフォルダの役割を果たす。102は“ContainmentRelationship”オブジェクトであり、Containerオブジェクトと、それが格納する対象となるオブジェクトを結び付ける役割を果たす。このオブジェクトは、例えば「格納者」「格納日」というような格納行為自体に係る情報を保持する。

【0004】 103は“Document Version”オブジェクトであり、DMA文書モデルにおける個々の文書オブジェクトに相当するものである。104の“Rendition”オブジェクトは、文書のある単一の表現形式のデータを管理するものである。Document Versionは、複数のRenditionを管理することによって、同一文書の複数の表現形式のデータを一括して管理することができる。105の“TransferElement”オブジェクトは、文書のある表現形式で表現した際の個々の構成要素を管理するためのオブジェクトであり、各構成要素と1:1の関係で存在し、対応する構成要素の実データへのアクセスを司る。

【0005】 次に、個々のオブジェクトの主要なプロパティを説明する。

【0006】 図2に、Container (101) の主要なプロパティを示す。Childrenプロパティ(201)は、Containerが他のContainerやDocVersionを格納する際に、Containerと格納対象オブジェクトを結び付けるContainmentRelationshipオブジェクトのリストを保持するためのプロパティである。アプリケーションは、このプロパティを参照することによって、あるContainerの格納するオブジェクトにアクセスすることができる。またParentプロパティ(202)は、このContainerオブジェクトをさらに格納するContainerオブジェクトが存在する場合に、その両者を結び付けるContainmentRelationshipを指すプロパティである。

【0007】 図3に、ContainmentRelationship (102) の主要なプロパティを示す。このオブジェクトは、格納するオブジェクトと格納対象オブジェクトとを結び付けるオブジェクトであるが、Headプロパティ(301)はその中の格納対象オブジェクトを指すプロパティである。また

Tailプロパティ (302) は、Headプロパティの指すオブジェクトを格納するContainerオブジェクトを指すプロパティである。

【0008】図4に、Document Version (103) の主要なプロパティを示す。Parentプロパティ (401) は、このDocument Versionと、このDocument Versionを格納するContainerとを結びつけるContainmentRelationshipを指すプロパティである。またRenditions Presentプロパティ (402) は、このDocument Versionにどのような表現形式が存在するかを示す文字列のリストである。個々の文字列はそれぞれRenditionに対応する。

【0009】図5に、Rendition (104) の主要なプロパティを示す。Rendition Typeプロパティ (501) は、このRenditionが管理する表現形式を示す文字列である。またTransfer Elements Presentプロパティ (502) は、このRenditionにどのような構成要素が存在するかを示す文字列のリストである。個々の文字列はそれぞれTransfer Elementに対応する。

【0010】図6に、Transfer Element (105) の主要なプロパティを示す。Reference Typeプロパティ (601) は、Referenceプロパティ (602) の内容の意味や文法を規定する整数コードである。具体的には、様々なファイルシステムのパス名やDMAオブジェクトのオブジェクトIDを示す整数コードがこのプロパティの値になる。またReferenceプロパティ (602) は、内容データを格納する際にそのデータが存在していた位置へのリファレンスを保持するための文字列形式のプロパティである。Component Typeプロパティ (603) は、このTransfer Elementが管理する内容データの、あるRenditionの管理するデータ全体における位置付けを示す文字列であり、どのような構成要素に相当するかを表す。Transfer Elementが、格納したデータをどのような形で管理するかについては、DMAでは規定されていない。

【0011】以上がDMA文書モデルに現れるオブジェクトの概略であるが、DMAの規定するモデルには、これらのオブジェクトを全て包含する論理的な「文書空間」という概念に対応する「文書空間オブジェクト」が存在する。例えば、アプリケーションがあるオブジェクトのオブジェクトIDを知っている場合には、そのIDを引数として文書空間オブジェクトのConnectObjectメソッドを呼ぶことにより、IDに該当するオブジェクトにアクセスすることができるようになる。

【0012】次に、文書の内容管理に関する例を詳細に示す(図7参照)。この図においては、各オブジェクトは文書の内容管理の仕組みに関するもののみ提示する。図中の矩形はオブジェクトの一つのインスタンスを表し、矩形内に含まれる情報は、そのオブジェクトのプロパティを表す。701は、Document Version (103) のインスタンスであり、Renditions Presentプロパティによって、このDocument VersionがWord、HTML、PDFの三種類の形式の

データを管理していることを表現している。この各表現形式に対応するRenditionオブジェクト (104) を得るためには、Document Versionに対してGetRenditionメソッド (702) を発行する。このメソッドの引数として、該当するRenditionがDocument VersionのRenditions Presentプロパティにおける何番目の文字列に対応するかを表す整数を付与することによって、望むRenditionオブジェクトを得る。

【0013】703, 704, 705は、それぞれWord形式、HTML形式、PDF形式のそれぞれの表現形式に対応するRenditionオブジェクトのインスタンスである。Renditionオブジェクトは、自らがどの表現形式に対応しているのかを示すRendition Typeプロパティと、その表現形式における文書の構成要素の各々を指し示す文字列の集合であるTransfer Elements Presentプロパティとを保持する。この各構成要素に対応するTransfer Elementオブジェクト (105) を得るためには、Renditionに対してGetTransferElementメソッド (706) を発行する。このメソッドの引数として、該当するTransfer ElementがRenditionのTransfer Elements Presentプロパティにおける何番目の文字列に対応するかを表す整数を付与することによって、望むTransfer Elementオブジェクトを得る。

【0014】707~709、710~712、713はTransfer Element (105) のインスタンスであり、それぞれWord形式、HTML形式、PDF形式の各表現形式を構成する個々の構成要素に対応している。Transfer Elementは、オリジナルのデータへのリファレンスを示すReferenceプロパティと、そのReferenceのタイプを表すReferenceTypeプロパティと、この構成要素が文書全体においてどのような位置付けにあるのかを示すComponent Typeプロパティとを保持する。

【0015】Transfer Element (105) に対応する個々の構成要素に対してどのような形でアクセスするかは、Transfer Elementにデータの実体の取得を依頼する際にどのメソッドを発行するかによって異なる。

【0016】1. ファイルとしてデータを取得したい場合には、Transfer Elementに対してGetFileAccessメソッド (714) を発行することで、File Access Object (715) を得る。このオブジェクトに対してCopyToメソッド (716) を発行することにより、データをアプリケーションプログラムが閲覧・編集可能なローカルファイルとしてコピーすることができる。

【0017】2. データのサイズが大規模であるために、その一部に対してSeekやReadによってアクセスしたい場合には、Transfer Elementに対してGetStreamメソッド (717) を発行することで、Stream Access Object (718) を得る。このオブジェクトに対してSeekやReadメソッド (719) を発行することにより、バイト単位でのデータの操作が実現できる。

【0018】3. データへのReference (URLやオブジェク

トIDなど)を取得したい場合には、Transfer Elementに対してGetReferenceメソッド(720)を発行することで、データにアクセスするためのリファレンスをそのものを得ることができる。このメソッドの引数として「ファイルパスを示すコード」や「オブジェクトIDを示すコード」を与えることによって、Transfer Elementが提供できる形式のデータへのリファレンスを得ることができる。例えば、これによって得られるDMAオブジェクトのオブジェクトIDを手がかりとして、オブジェクトIDに対応するDMAオブジェクトにアクセスすることができる。

【0019】次に、文書の論理的な構造を明示的に表す情報を含む文書である「構造化文書」について説明する。

【0020】構造化文書の形式の一つに、論理構造を明示的に表す情報をテキスト中に埋め込む方法がある。一般にユーザが作成した構造化文書(以下「文書インスタンス」と呼ぶ)は、文書の論理構造を規定する論理構造定義部と、文書の内容を表す内容テキスト部からなることが多い。論理構造定義には、その文書の論理構造と、その構成要素を表すマーク(以下「タグ」と呼ぶ)が定義される。また内容テキスト部には、論理構造定義部で定義されたタグを、そのタグに対応する論理構造の内容となる文字列が一意に定まるように挿入し、文書の論理構造を明示的に表現する。

【0021】このようにして構造化された文書インスタンスをレイアウトして紙面や画面に出力する際には、論理構造の各構成要素(以下「エレメント」と呼ぶ)をどのような書式で出力するかを規定する出力書式定義を記述したファイルを参照し、出力すべきイメージを生成する。この方法によると、文書インスタンスと出力書式定義とが独立しているため、出力に用いる個々の装置やシステムにかかわらず文書インスタンスを交換することができる。

【0022】また、こうした構造化文書における文字列の内容は、例えば「<著者名>」や「<タイトル>」というような、エレメントと一対一で対応するタグの挿入によって明示的に表現されるため、構造化文書に対応した全文検索システム等のツールと組み合わせることにより、文書インスタンスの集合をそのままデータベースとして利用することができる。構造化文書とその利用の詳細については、例えば文献「SGMLのススメ」(吉岡誠編著、オーム社)等において詳しく解説されている。

【0023】こうした利点から、大量の文書を蓄積、利用する文書処理システムにおける文書管理形式として、構造化文書形式の採用が進んでいる。また、構造化文書を単なるテキストデータとして管理するのではなく、文書の個々の構成要素をオブジェクトによって表現し、文書全体をオブジェクトのツリー構造の形で管理することにより、個々の構成要素に対するアクセス権等、元々の文書内には記述されていないプロパティを付加したり、

構成要素の再利用性を高めることを狙った文書管理システムも開発されている。

【0024】図8に、構造化文書の国際標準(ISO-8879)であるSGML形式で記述した文書の例を示す。801はファイル“特許.dtd”に記述された論理構造定義の例である。802は、「特許」というエレメントが内容データとして「明細書」「図面」という二つのエレメントを持つことを示す。また803は、「明細書」というエレメントがさらに内容データとして「発明名称」「請求範囲」「詳細説明」などのエレメントを持つことを示す。804は、「図」というエレメントが“entity”という名前の属性を持ち、その属性値が、ある名前を付けられたデータを参照するためのエンティティ参照であることを示している。

【0025】805は、801の論理構造定義に沿った文書の例である。807は、論理構造定義としてファイル“特許.dtd”に記述された定義を用いること、及びこの文書全体を包含するエレメントが「特許」というエレメントであることを表している。

【0026】806は、実際に文書の論理構造を記述した文書インスタンスである。809と810は、それぞれ「明細書」というエレメントの開始と終了を表すタグであり、このタグで挟まれた部分がエレメント「明細書」の内容に相当することを示している。曖昧さがない場合、一部のタグが省略されることもある。

【0027】また811は、「図」というエレメントが“entity”という名前の属性を持ち、かつその属性値が「Fig1」という参照名の実体(エンティティ)を参照していることを示している。そして808は、「Fig1」という参照名のエンティティが非SGML形式の外部データであり、それが外部ファイル“図1.bmp”として存在することを示す実体宣言である。このように、バイナリデータなどの非SGMLデータは、外部データの所在と種類、およびSGML文書内でそれを参照するための参照名をエンティティ宣言によって定義し、その参照名をエレメントの属性の値とすることによって、SGML文書と関連付けられる。

【0028】図8の文書インスタンスの構造をツリー状に表現したものを図9に示す。

【0029】

【発明が解決しようとする課題】DMAで提案されている文書モデル(以下「DMA文書モデル」と略す)においては、複数の構成要素からなる文書の内容を表現するための枠組みとして、図7に示したように一つの文書が複数のファイルから構成されるような、構成要素がフラットに並ぶ構造を表現するための方法が提案されている。しかし、SGMLのように、構成要素がツリー構造を構成するような構造化文書を表現するための枠組みは提案されていない。

【0030】DMA文書モデルにおいて、既存のDMA仕様のフレームワークを損なうことなく構造化文書を管理する

のが困難であるのは、次のような理由による。

【0031】例えばSGML文書において、Transfer Elementを文書の構成要素と1:1で対応させるには、次の二つの方式が考えられる。

【0032】方式1：構造化文書の論理構造の要素に1:1でTransfer Elementを割り当てる。

【0033】方式2：文書インスタンス(ソースとなるテキスト)全体を、DTDやSGML宣言などと同様に一つの構成要素とみなし、単一のTransferElementを割り当てる。

【0034】図10は、方式1に対応するものであり、「構造化文書の構成要素 = エレメント」とみなして、個々のエレメントに対してTransfer Elementを割り当てた場合の例を表したものである。この場合の問題点としては、要素数が多くなると一つのRenditionが管理するTransfer Elementが膨大な数になる点だけでなく、「あるエレメントが複数のエレメントから構成される」というようなエレメント間の包含関係によるツリー構造が表現できていない、という点も挙げられる。

【0035】図11は、方式2に対応するものであり、文書インスタンス(ソースとなるテキスト)全体を、DTDやSGML宣言などと同様に一つの構成要素とみなしてTransferElementを割り当てた例である。この場合には、構造化文書の構造を閲覧しようとする際にはソースをチェックアウトし、構造化文書エディタ等のツールを介して閲覧する必要がある。

【0036】

【課題を解決するための手段】本発明では上記課題を解決するため、エレメントやその内容文字列等をDMAベークラス(DMAに現れるクラスの全てのスーパークラス)のサブクラスとして定義する。そして、構造化文書の文書インスタンスのツリー構造は、エレメントやエンティティ、及び文字列データを表すオブジェクトのツリー構造によって表現する。

【0037】さらに、構造化文書を表現するオブジェクトのツリー構造の根(トップノード)に相当するオブジェクトを、「文書の内部構造」を表す単一のTransferElementから参照できるようにする。具体的には、このTransfer Elementに文書インスタンスのツリー構造のトップノードに相当するオブジェクトのIDを持たせることにより、「GetReference」メソッドによってそのオブジェクトIDを得られるようにする。

【0038】このような構成によって、上記の課題が以下のように解決される。まず、構造化文書に現れる諸概念をオブジェクトとして表現し、文書の論理構造をそのオブジェクト群のツリー構造として表現することにより、アプリケーションはSGMLパーサを用いることなくツリー型の論理構造を参照することができる。

【0039】また、「文書の内部構造」を表す単一のTransfer Elementを介して論理構造の根となるエレメントを参照できるようにすることで、エレメントの数に応じ

てTransfer Elementの数も増大するという問題は解決される。

【0040】さらに、論理構造を表現するオブジェクトはDMAベークラスのサブクラスのインスタンスであるため、DMAで規定されたアプリケーション・インタフェースを介して構造化文書のツリー構造を参照することができるという利点もある。

【0041】

【発明の実施の形態】ここでは、構造化文書形式の例としてSGML形式を取り上げ、SGML形式の文書をDMAモデルに組み込むための拡張DMAモデルの例を示す。

【0042】まず、本発明を実現するための装置構成例を図12に示す。1201は入出力装置であり、利用者からの文書データ処理要求を受け付け、またその処理結果を利用者に提示するための装置である。1202は、実際に文書処理を実行する制御装置である。1203は、文書データを格納するための外部記憶装置である。

【0043】制御装置1202において、制御部1204は文書処理に関する一連の処理を制御する。具体的には、入出力装置から入力された文書データ処理要求に従って、文書処理部1205の保持する手続きを内部メモリ1208に展開して文書処理を実行し、文書処理の過程で必要になる文書データを外部記憶装置1203から取り出し、要求される処理結果を入出力装置1201に出力するまでの処理を制御する。

【0044】文書処理部1205は、文書理用手続き1206と文書管理手続き1207を保持する。文書管理手続き1207は、例えばファイルシステムやリレーショナルデータベースによって実現される外部記憶装置1203に格納される文書データを、メソッドを持つオブジェクト群として文書利用手続き1206に提供する手続きであり、文書の構成情報の管理や内容データへのアクセスの管理、変更履歴の管理等の機能を提供するものである。文書処理手続き1206は、文書管理手続き1207がオブジェクトとして提供する文書データを利用して、文書の閲覧、編集、変換、配布といった処理を行う手続きであり、文書管理手続き1207の機能を利用するアプリケーションプログラムという位置付けになる。

【0045】本発明は、文書管理手続き1207に関するものであり、構造化文書であるSGML文書を構成要素単位のオブジェクト群として文書利用手続き1206に提供する際の、オブジェクトのクラス構成とその利用方法に関する発明である。

【0046】次に、オブジェクトによってSGML文書の論理構造を表現する方法について説明する。SGML文書の内容をDMAのアプリケーションインタフェースを介して操作できるようにするために、SGML文書の表現に用いるオブジェクトのクラスをDMAオブジェクトのサブクラスとして定義する。図13に、SGML文書の表現に用いるクラスとその間のクラス階層を示す。図中では、矢印の元がス

ーパークラスを、矢印の先がサブクラスを表す。

【0047】1301はDMAで規定されたDMAベースクラスであり、DMAのアプリケーションインタフェースを介してアクセスできる全てのDMAオブジェクトがこのサブクラスである。このクラスのオブジェクトは「プロパティ」としてオブジェクトID等の様々なデータを保持するが、そのプロパティは、このDMAクラスがサポートするIdmaPropertiesインタフェースのメソッドを用いて参照することができる。図13中のクラスは全てDMAクラスのサブクラスであるため、そのインスタンスのプロパティはIdmaPropertiesインタフェースのメソッドによって参照できることになる。

【0048】1302のSGML Baseクラスは、DMAクラスのサブクラスであり、SGML関連の全てのクラスのスーパークラスとなる仮想的なクラスである。あるオブジェクトが特定のクラスに属する否かを調べるためのIsOfClassメソッド(DMAベースクラスのメソッド)を用いてそれがSGML Baseクラスのインスタンスであることが判明すれば、そのオブジェクトがSGML文書を表現するために導入されたオブジェクトであることがわかる。

【0049】1303のNonSgmlDataクラスは、SGML Baseクラスのサブクラスであり、バイナリ形式のデータ等の非SGMLデータを格納するためのクラスである。図14に、NonSgmlDataの主なプロパティを示す。1401のNotationは、データの記法を示す文字列を保持するプロパティである。1402のBinaryDataは、バイナリ形式の実データを保持するためのプロパティである。

【0050】1304のSGML宣言クラスは、SGML Baseクラスのサブクラスであり、SGML宣言を格納するためのクラスである。SGML宣言とは、あるSGML文書がどのような方法で記述されているかを述べたものであり、文書を記述するための文字集合や制御コードなどがこれに含まれる。図15に、SGML宣言オブジェクトの主なプロパティを示す。1501のNameは、このSGML宣言に付与する名称を保持する文字列形式のプロパティである。1502のContentプロパティは、テキスト形式のSGML宣言の実データを保持するための文字列形式のプロパティである。

【0051】1305のDTDクラスは、SGML Baseクラスのサブクラスであり、SGMLにおける論理構造定義であるDTD(Document Type Definition)を格納するためのクラスである。図16に、DTDオブジェクトの主なプロパティを示す。1601のNameは、このDTDに付与する名称を保持する文字列形式のプロパティである。1602のContentプロパティは、テキスト形式のDTDの実データを保持するための文字列形式のプロパティである。1603のFormatInfは、このDTDに沿った文書に適用できる可視化情報(1307)のリストを保持するためのオブジェクトである。

【0052】1306のSgmlNodeクラスは、SGML Baseクラスのサブクラスである。“SgmlNode”はSGML文書の論理的なツリー構造を構成するノード・オブジェクトの総称で

あり、このクラスはそれらのクラス群のスーパークラスとなる仮想的なクラスである。実際のインスタンスは、サブクラスであるElementNode, EntityNode, 及びDataNodeのいずれかのインスタンスになる。

【0053】1307の可視化情報クラスは、SGML Baseクラスのサブクラスであり、SGML文書を画面や紙面に出力する際に必要になる可視化情報を格納するためのクラスである。図17に、可視化情報オブジェクトの主なプロパティを示す。1701のNameは、この可視化情報に付与する名称を保持する文字列形式のプロパティである。1702のContentプロパティは、テキスト形式の可視化情報の実データを保持するための文字列形式のプロパティである。

【0054】1308のAttributeクラスは、SGML Baseクラスのサブクラスであり、エレメントの個々の属性(Attribute)を管理するためのクラスである。属性名と宣言値、及び属性値を保持する。図18に、Attributeオブジェクトの主要なプロパティを示す。1801の属性名は、この属性(Attribute)の名称を示す文字列を保持するためのプロパティである。1802の宣言値は、属性値の種類(型に相当する情報)を規定する文字列を保持するためのプロパティである。1803の属性値は、属性値の実データを保持するオブジェクトである。具体的には、文字列データの場合にはDataNodeになり、バイナリデータの場合にはNonSgmlDataになる。

【0055】1309のElementNodeクラスは、SgmlNodeクラスのサブクラスであり、SGML文書のエレメントの情報を管理するためのクラスである。このオブジェクトがエレメントの内容データとしてSgmlNodeのリストを保持することにより、文書の論理的なツリー構造が構築される。また、エレメントの属性をAttributeオブジェクトのリストの形で提供する。図19に、ElementNodeオブジェクトの主要なプロパティを示す。1901のNameは、このエレメントの名称を保持する文字列形式のプロパティである。1902のContentsは、エレメントの内容データを表現するSgmlNodeオブジェクトのリストを保持するためのプロパティである。具体的には、ElementNode、DataNode、及びEntityNodeの任意の並びになる。1903のAttributesは、エレメントの属性を表現するAttributeオブジェクトのリストを保持するためのプロパティである。

【0056】1310のEntityNodeクラスは、SgmlNodeクラスのサブクラスであり、SGML文書における一般実体参照を表現するためのクラスである。入力されるSGML文書の中で、SGMLデータの実体参照はSgmlNodeに展開し、非SGMLデータへの実体参照のみをEntityNodeによって管理する。図20に、EntityNodeオブジェクトの主要なプロパティを示す。2001のNameは、このエンティティ参照に付与する名称を示す文字列形式のプロパティである。2002のNonSgmlDataは、このエンティティ参照の参照対象となる非SGMLデータを保持するNonSgmlDataオブジェクトを

保持するためのプロパティである。

【0057】1311のDataNodeクラスは、SgmlNodeクラスのサブクラスであり、SGML文書の論理的なツリー構造の葉に相当するテキストデータを管理するためのクラスである。図21に、DataNodeオブジェクトの主要なプロパティを示す。2101のTypeは、保持する文字列データの種類(ex. CDATA, SDATAなど)を表現するための文字列形式のプロパティである。2102のTextDataは、このDataNodeオブジェクトが保持する文字列形式の実データを保持するためのプロパティである。

【0058】図22に、ElementNodeオブジェクトとAttributeオブジェクトの構成を詳細に示す。ElementNode(2201)は、SGML文書のエレメントの情報を管理するオブジェクトである。このオブジェクトが、エレメントの内容データを示すContentsプロパティ(2202)の値として子ノード(ElementNodeを含むSgmlNode)のリスト(2204)を保持することにより、文書の論理的なツリー構造が構成される。また、ElementNodeはエレメントの属性としてAttributeオブジェクトのリストを値とするAttributesプロパティ(2203)を持つ。Attributeオブジェクト(2205, 2206, 2207)は、少なくとも属性名、宣言値及び属性値の三つのデータを保持するオブジェクトである。属性値は、宣言値(属性値の「型」に相当するデータ)がEntityの場合にはEntityNode(2209)となり、その他の場合にはテキストデータを管理するDataNode(2208, 2211)となる。EntityNode(2209)は、NonSgmlDataオブジェクト(2210)をプロパティとして持つことにより、SGML文書内での非SGMLデータへの実体参照を表現する。

【0059】図23に、図8に示したSGML文書の論理構造を図13のオブジェクトを用いて表現した例を示す。文書の論理構造は、エレメント「特許」に対応するElementNode(2301)を根とするSgmlNodeのツリー構造によって表現される。2301のElementNodeは、エレメント「明細書」に対応するElementNode(2302)と、エレメント「図面」に対応するElementNode(2303)を下位要素とする。また、2302と2303のElementは、さらに下位要素としてElementNodeなどのSgmlNodeの並びを持つことができる。このように、ElementNodeが下位要素として再帰的にElementNodeを持つことによって、ツリー状の構造を構成する。

【0060】文字列に対応するエレメントは、2304のElementNodeと2305のDataNodeに示すように、内容文字列を保持するDataNode(2305)をElementNode(2304)の下位要素とする形で表現する。

【0061】非SGMLデータのエンティティ参照は、エレメントの属性を介して表現されることになるが、これを本拡張DMA文書モデルにおいては2306~2309に示す形で表現する。具体的には、エレメントの個々の属性を表現するAttributeオブジェクトのリストをElementNodeが保持しており、その中のあるAttribute(2307)が、エン

ティティを参照するための属性を表現している。2307のAttributeの宣言値が“ENTITY”であることが、このAttributeの属性値の種類がEntityNodeであることを示す。そして、属性値であるEntityNode(2308)においてエンティティ参照の対象を指すNonSgmlDataプロパティの値が、元来“図1.bmp”という外部ファイルに格納されていたデータを格納するためのオブジェクトNonSgmlData(2309)になる。

【0062】また、本拡張モデルにおいては、Transfer Elementを図24に示すように実装する。つまり、DMAで規定されているプロパティの他に、このTransfer Elementが管理する内容データの文書管理システム内部におけるオブジェクトID(01ID)を管理するためのプロパティであるTarget01IDプロパティ(2401)を実装する。

【0063】図25に、図23に示した論理構造を含むSGML文書全体を拡張DMA文書モデルによって表現した例を示す。2501~2507は全てDMAモデルで規定されているオブジェクトである。2501は「特願平xx-xxxx」という特許の文書を表すDocVerionオブジェクトである。2502~2504は、その特許をそれぞれHTML形式、SGML形式、JPEG形式で表現したデータを管理するためのRenditionオブジェクトである。2505~2508は、SGML形式の文書を構成する個々の構成要素へのアクセスを司るTransfer Elementオブジェクトである。2505及び2507は、SGML宣言、及びDTDにアクセスするためのTransfer Elementであり、それぞれSGML宣言を格納するためのSGML宣言オブジェクト2508、DTDを格納するためのDTDオブジェクト2509のオブジェクトIDをTarget01IDプロパティ(2401)によって保持している。またDTDオブジェクト2509からは、FormatInfプロパティ(1603)を参照することによって、そのDTDに沿った文書に適用できる可視化情報(2510)にアクセスすることができる。

【0064】2506は、SGML文書の論理構造にアクセスするためのTransfer Elementであり、論理構造を表すツリー構造の根となるElementNodeオブジェクト(2511)へのアクセスをアプリケーションに提供する。2513に、2512のTransfer ElementとElementNodeの関係をより具体的に示す。2514と2515は、それぞれ2506と2511の内容を詳細に示したものである。Transfer Element(2514)ではTarget01IDプロパティの値としては、論理構造の根となるElementNode(2516)のオブジェクトIDが格納される。

【0065】図26に、アプリケーションがTransfer Element(2515)にアクセスしてからElementNode(2516)にアクセスするまでの流れを示す。アプリケーションは、Transfer Elementにアクセスすると(2601)、そのTransfer ElementのGetReferenceメソッドを「オブジェクトIDを示すコード」を引数にして呼ぶ(2602)。そして、Transfer ElementのTarget01IDプロパティの値である「SGML文書の論理構造の根となるノードのオブジェクトID」を得る(2603)。次に、そのオブジェクトIDを引数として、文書

空間オブジェクトのConnectObjectメソッドを呼ぶことにより(2604)、該当するElementNodeを得る(2605)。

【0066】この手続きは、2505及び2507のTransfer Elementから、2508及び2509のそれぞれのオブジェクトにアクセスする際にも同様に用いられる。

【0067】論理構造を表現するために本拡張文書モデルにおいて導入したオブジェクトは全てDMAオブジェクトのサブクラスであり、DMAオブジェクトがサポートするメソッドも同じようにサポートすることになる。そのため、各オブジェクトはDMAオブジェクトがサポートする「オブジェクトのプロパティを参照するためのメソッド群」をサポートする。これにより、論理構造の根となるノードへのアクセスを得たアプリケーションはそれらのメソッド群を利用することによって、ElementNodeの下位要素を参照したり、ElementNodeの属性を表すAttributeを参照したり、テキストデータを保持するDataNodeの内容を参照したりすることができる。これにより、DMAで規定されているメソッド群を用いて、文書の論理構造全体を参照することができる。

【0068】図27に、本拡張モデルにおいてアプリケーションが論理構造を持った文書の内容を参照するための処理手順を示す。アプリケーションは、まず一つの文書に相当するDocVersionを取得する(2701)。DMA文書モデルにおいてこれを取得する方法としては、検索によって得る方法や、オブジェクトIDが分かっているならばそれを引数としてDocSpaceオブジェクトのConnectObjectメソッドを呼ぶ方法が用意されている。

【0069】DocVersionを取得すると、次にそのオブジェクトのRenditions Presentプロパティを取得する(2702)。このプロパティは文字列のリストになっており、各文字列はそれぞれこのDocVersionが管理するRenditionを表現している。アプリケーションは、このリストの中で望むRenditionに該当する文字列を選択してその文字列のリストにおける位置を取得し(2703)、その位置を引数としてDocVersionのGetRenditionメソッドを呼ぶ。その結果、該当するRenditionオブジェクトが得られる(2704)。

【0070】Renditionを取得すると、次にそのオブジェクトのTransfer Elements Presentプロパティを取得する(2705)。このプロパティは文字列のリストになっており、各文字列はそれぞれこのRenditionが管理するTransfer Elementを表現している。本拡張モデルにおいては、SGML文書を表現するためにSGML宣言、論理構造及びDTDに対応する3つのTransfer Elementを用意しており、それぞれを表現する文字列がTransfer Elements Presentプロパティの値として設定されている。アプリケーションは、このリストの中で望むTransfer Elementに該当する文字列を選択してその文字列のリストにおける位置を取得し(2706)、その位置を引数としてRenditionのGetTransferElementメソッドを呼ぶ。その結果、該当するT

ransfer Elementオブジェクトが得られる(2707)。

【0071】Transfer Elementを取得すると、次にそのTransfer Elementが指し示すオブジェクトを取得する。そのために、まずTransfer ElementのGetReferenceメソッドを「ObjectIDを示すコード」を引数として呼び、そのTransfer Elementが指し示すオブジェクトのObjectIDを得る(2708)。さらに、そのObjectIDを引数としてDocSpaceオブジェクトのConnectObjectメソッドを呼ぶことにより、ObjectIDに対応するオブジェクトを得る(2709)。

【0072】2709において取得したオブジェクトの種類がSGML宣言かDTDのいずれかである場合(2710)には、そのオブジェクトのContentプロパティを取得することによって、内容を得ることができる(2711)。

【0073】2709において取得したオブジェクトの種類がElementNodeである場合(つまり「論理構造」に対応するTransfer Elementを選択した場合)には、そのエレメントに関するどのような情報を得たいかによって処理が異なる(2712)。エレメントの下位要素にアクセスする場合には、ElementNodeのContentsプロパティを取得する(2713)。このプロパティの値は、エレメントの下位要素に相当するSgmlNodeのリストになっているため、そのリストからアクセスしたいSgmlNodeを取得する(2714)。そのSgmlNodeの種類がElementNodeである場合(2715)には、再帰的に2712の処理に戻り、そのElementNodeの内容を参照する。また、2715においてSgmlNodeの種類がDataNodeである場合には、そのオブジェクトのTextDataプロパティを取得することにより、内容文字列を得ることができる(2716)。

【0074】2712～2716は、エレメントの下位要素にアクセスする処理であるが、エレメントのAttributeにアクセスするにはElementNodeのAttributesプロパティを取得する(2717)。このプロパティの値はAttributeオブジェクトになっているため、そのリストからアクセスしたいAttributeを取得する(2718)。さらに取得したAttributeから望むプロパティを取得する(2719)。このとき、取得するプロパティがAttributeの属性値プロパティである場合には、その値はDataNodeオブジェクトかEntityNodeオブジェクトになる(2720)。属性値プロパティの値がDataNodeオブジェクトである場合には、TextDataプロパティを取得することにより、内容文字列を得ることができる。また属性値プロパティの値がEntityNodeオブジェクトである場合には、EntityNodeのNonSgmlDataプロパティの値であるNonSgmlDataオブジェクトを取得し(2721)、さらにそのオブジェクトのBinaryDataプロパティの値を取得することによって、エンティティ参照の対象となる非SGMLデータを参照することができる。

【0075】以上、図27に示す処理フローによって、アプリケーションは文書の内容を論理構造のツリーに沿って閲覧していくことができる。

【0076】また、SGMLにおいて、特定のアプリケーション固有の処理命令を表すPIについても表現する場合には、図28に示すようなPIクラスを導入すればよい。2801は、PIの内容を表す文字列型のデータを保持するContentプロパティである。このPIは、最上位に位置するエレメントと同じ階層に現れうるため、論理構造のツリー構造のトップになるノードはエレメントではなく、最上位エレメントとPIとを保持するノードとなる。このノードは、例えば図29に示すDocNodeクラスによって実現できる。このノードは、DTDプロパティ2901とSGML宣言プロパティ2902で、それぞれDTDオブジェクトとSGML宣言プロパティのオブジェクトID(01ID)を保持している。この01IDを引数としてDocSpaceオブジェクトのConnectObjectプロパティを呼ぶことにより、DTDオブジェクトやSGML宣言オブジェクトにアクセスすることができる。つまり、このDocNodeを用いれば、DTDやSGML宣言にアクセスするためのTransfer Elementは必要なく、このDocNodeにアクセスするためのTransfer Elementのみ用意すればよい。

【0077】DocNodeへのアクセスするTransfer Elementのみを用意した場合の文書モデルの構成例を図30に示す。3001~3005は全てDMAモデルで規定されているオブジェクトである。3001は「特願平xx-xxxx」という特許の文書を表すDocVersionオブジェクトである。3002~3004は、その特許をそれぞれHTML形式、SGML形式、JPEG形式で表現したデータを管理するためのRenditionオブジェクトである。3005は、SGML形式の文書をツリー形式で表現するオブジェクト群へのアクセスを司るTransfer Elementオブジェクトであり、論理構造を表すツリー構造の根となるDocNodeオブジェクト(3006)へのアクセスをアプリケーションに提供する。3008に、3007のTransfer ElementとElementNodeの関係をより具体的に示す。3009と3010は、それぞれ3005と3006の内容を詳細に示したものである。Transfer Element(3009)ではTarget01IDプロパティの値としては、論理構造の根となるElementNode(3010)のオブジェクトIDが格納される。Transfer Elementに対して、「オブジェクトIDを示すコード」を引数としてGetReferenceメソッドを呼ぶことにより、アプリケーションはDocNodeの01IDを得ることができる。この01IDを引数としてDocSpaceオブジェクトのConnectObjectメソッドを呼ぶことにより、アプリケーションはDocNode(3006)にアクセスすることができる。

【0078】DocNodeオブジェクト3006からは、01IDを介してDTDオブジェクト3011とSGML宣言オブジェクト3013にアクセスすることができ、さらにDTDオブジェクトからはFormatInfプロパティ(1603)を介して可視化情報3012にアクセスすることができる。また、DocNodeはContentsプロパティ(2903)において、文書の最上位エレメント(3015)と任意個のPI(例:3014)を保持する。最上位エレメント以下の構造は図25におけるモデルと全く同様であ

る。

【0079】以上、図30、図28及び図29に示したオブジェクト構成によっても、本発明を実現することができる。

【0080】

【発明の効果】以上のように、本発明によれば、構造化文書を構成するエレメントやそれに付随する内容文字列、Attribute等をDMAベースクラスのサブクラスとして定義したオブジェクトで表現し、文書インスタンスをそれらのオブジェクトのツリー構造によって表し、さらにそのツリー構造のトップノードを「文書の内部構造」を表す単一のTransfer Elementから参照できるようにすることによって、既存のDMA仕様準拠の文書データへのアクセスと、構造化文書の内部構造へのアクセスとを、DMAで規定されたインタフェースという統一的なアクセス方法によって実現することができる。

【図面の簡単な説明】

【図1】DMAにおいて規定されている文書モデルの概要を示す図である。

【図2】Containerオブジェクトの主なプロパティを示す図である。

【図3】ContainmentRelationshipオブジェクトの主なプロパティを示す図である。

【図4】Document Versionオブジェクトの主なプロパティを示す図である。

【図5】Renditionオブジェクトの主なプロパティを示す図である。

【図6】Transfer Elementオブジェクトの主なプロパティを示す図である。

【図7】DMA文書モデルにおける文書の内容データ管理の例を表す図である。

【図8】SGML形式の文書の具体例を示す図である。

【図9】SGML文書の論理的な構造をツリー形式で表現した例を表す図である。

【図10】SGML文書をDMA文書モデルで表現しようとする際に、個々のエレメントにTransfer Elementを割り当てて表現する例を示す図である。

【図11】SGML文書をDMA文書モデルで表現しようとする際に、文書インスタンス全体を一つの構成要素とみなして表現する例を示す図である。

【図12】本発明を実現するための装置構成例を示す図である。

【図13】SGML文書を表現するために導入するオブジェクトのクラスを示す図である。

【図14】NonSgmlDataオブジェクトの主なプロパティを示す図である。

【図15】SGML宣言オブジェクトの主なプロパティを示す図である。

【図16】DTDオブジェクトの主なプロパティを示す図である。

【図17】可視化情報オブジェクトの主なプロパティを示す図である。

【図18】Attributeオブジェクトの主なプロパティを示す図である。

【図19】ElementNodeオブジェクトの主なプロパティを示す図である。

【図20】EntityNodeオブジェクトの主なプロパティを示す図である。

【図21】DataNodeオブジェクトの主なプロパティを示す図である。

【図22】ElementNodeオブジェクトとAttributeオブジェクトの実例をより詳細に示した図である。

【図23】SGML文書の論理構造をオブジェクトによって表現した例を示す図である。

【図24】Transfer Elementを実装する際にはTargetOIDプロパティを付加する例を示す図である。

【図25】拡張DMA文書モデルによってSGML文書全体を表現した例を示す図である。

【図26】アプリケーションが文書の論理構造の根となるElementNodeにアクセスするための処理を示す図である。

【図27】本拡張モデルにおいてアプリケーションが論理構造を持った文書の内容を参照するための処理手順を示す図である。

【図28】PIクラスの主なプロパティを示す図である。

【図29】DocNodeクラスの主なプロパティを示す図である。

【図30】DocNodeを参照するTransfer Elementを用いた文書モデルの構成例を示す図である。

【符号の説明】

101…DMA文書モデルにおけるContainerオブジェクトである。
 102…DMA文書モデルにおけるContainment Relationshipオブジェクトである。
 103…DMA文書モデルにおけるDocument Versionオブジェクト、
 104…DMA文書モデルにおけるRenditionオブジェクト、
 105…DMA文書モデルにおけるTransferElementオブジェクト、
 201…ContainerオブジェクトのChildrenプロパティの仕様、
 202…ContainerオブジェクトのParentプロパティの仕様、
 301…ContainmentRelationshipオブジェクトのHeadプロパティの仕様、
 302…ContainmentRelationshipオブジェクトのTailプロパティの仕様、
 401…Document VersionオブジェクトのParentプロパティの仕様、
 402…Document VersionオブジェクトのRenditions Pres

entプロパティの仕様、

501…RenditionオブジェクトのRendition Typeプロパティの仕様、

502…RenditionオブジェクトのTransfer Elements Presentプロパティの仕様、

601…Transfer ElementオブジェクトのReference Typeプロパティの仕様、

602…Transfer ElementオブジェクトのReferenceプロパティの仕様、

10 603…Transfer ElementオブジェクトのComponent Typeプロパティの仕様、

701…Document Versionオブジェクトの例、

702…Document VersionオブジェクトのGetRenditionメソッド、

703…Word形式のデータに対応するRenditionオブジェクトの例、

704…HTML形式のデータに対応するRenditionオブジェクトの例、

20 705…PDF形式のデータに対応するRenditionオブジェクトの例、

706…RenditionオブジェクトのGetTransferElementメソッド、

707～709…Word形式のデータに対応するTransfer Elementオブジェクトの例、

710～712…HTML形式のデータに対応するTransfer Elementオブジェクトの例、

713…PDF形式のデータに対応するTransfer Elementオブジェクトの例、

30 714…Transfer ElementオブジェクトのGetFileAccessメソッド、

715…DMA文書モデルにおけるFile Access Object、

716…File Access ObjectのCopyToメソッド、

717…Transfer ElementオブジェクトのGetStreamメソッド、

718…DMA文書モデルにおけるStream Access Object、

719…Stream Access ObjectのSeekメソッド及びReadメソッド、

720…Transfer ElementオブジェクトのGetReferenceメソッド、

40 801…ファイル“特許.dtd”に記述された論理構造定義の例、

802…エレメント「特許」の定義例、

803…エレメント「明細書」の定義例、

804…エレメント「図」の定義例、

805…801の論理構造定義に沿った文書の例、

806…実際に文書の論理構造を記述した文書インスタンスの例、

807…Doctype宣言の例、

808…エンティティ宣言の例、

50 809…エレメント「明細書」の開始タグ、

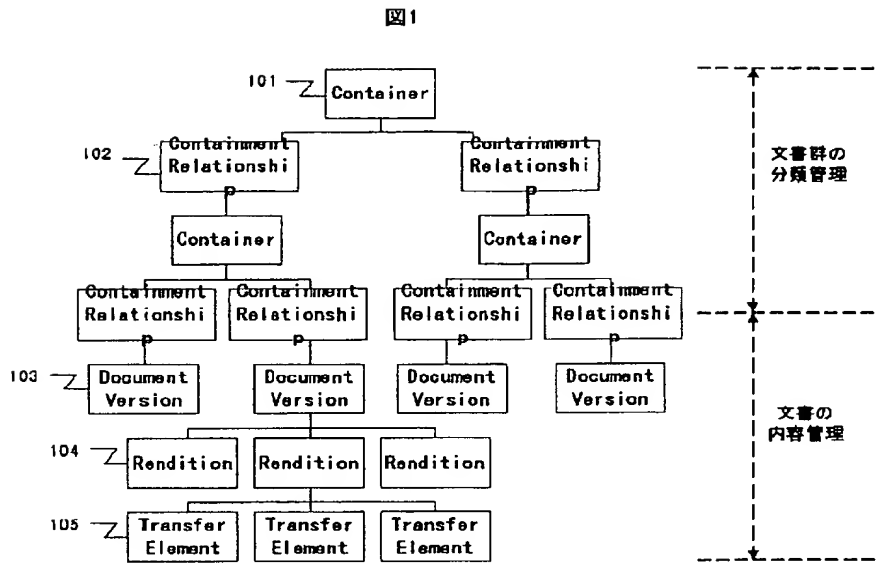
810…エレメント「明細書」の開始タグ、
 811…エレメント「図」の例、
 1201…利用者からの文書データ処理要求を受け、またその処理結果を利用者に提示するための入出力装置、
 1202…実際に文書処理を実行する制御装置、
 1203…文書データを格納するための外部記憶装置、
 1204…文書処理に関る一連の処理を制御する制御部、
 1205…文書処理用手続き1206と文書管理手続き1207を保持する文書処理部、
 1206…書管理手続き1207がオブジェクトとして提供する文書データを利用して、文書の閲覧、編集、変換、配布といった処理を行う文書利用手続き、
 1207…書の構成情報の管理や内容データへのアクセスの管理、変更履歴の管理等の機能を提供する文書管理手続き、
 1208…内部メモリ、
 1301…DMAで規定されたDMAベースクラス、
 1302…拡張文書モデルにおけるSGML Baseクラス、
 1303…拡張文書モデルにおけるNonSgmlDataクラス、
 1304…拡張文書モデルにおけるSGML宣言クラス、
 1305…拡張文書モデルにおけるDTDクラス、
 1306…拡張文書モデルにおけるSgmlNodeクラス、
 1307…拡張文書モデルにおける可視化情報クラス、
 1308…拡張文書モデルにおけるAttributeクラス、
 1309…拡張文書モデルにおけるElementNodeクラス、
 1310…拡張文書モデルにおけるEntityNodeクラス、
 1311…拡張文書モデルにおけるDataNodeクラス、
 1401…NonSgmlDataクラスのNotationプロパティの仕様、
 1402…NonSgmlDataクラスのBinaryDataプロパティの仕様、
 1501…SGML宣言クラスのNameプロパティの仕様、
 1502…SGML宣言クラスのContentプロパティの仕様、
 1601…DTDクラスのNameプロパティの仕様、
 1602…DTDクラスのContentプロパティの仕様、
 1603…DTDクラスのFormatInfプロパティの仕様、
 1701…可視化情報クラスのNameプロパティの仕様、
 1702…可視化情報クラスのContentプロパティの仕様、
 1801…Attributeオブジェクトの属性名プロパティの仕様、
 1802…Attributeオブジェクトの宣言値プロパティの仕様、
 1803…Attributeオブジェクトの属性値プロパティの仕様、
 1901…ElementNodeオブジェクトのNameプロパティの仕様、
 1902…ElementNodeオブジェクトのContentsプロパティの仕様、
 1903…ElementNodeオブジェクトのAttributesプロパティの仕様、

2001…EntityNodeオブジェクトのNameプロパティの仕様、
 2002…EntityNodeオブジェクトのNonSgmlDataプロパティの仕様、
 2101…DataNodeオブジェクトのTypeプロパティの仕様、
 2102…DataNodeオブジェクトのTextDataプロパティの仕様、
 2201…ElementNodeオブジェクトの例、
 2202…ElementNodeオブジェクトのContentsプロパティ、
 2203…ElementNodeオブジェクトのAttributesプロパティ、
 2204…ElementNodeオブジェクトのContentsプロパティの内容の例、
 2205～2207…Attributeオブジェクトの例、
 2208…Attributeオブジェクトの属性値に該当するDataNodeの例、
 2209…Attributeオブジェクトの属性値に該当するEntityNodeの例、
 2210…非SGML形式のデータを包含するNonSgmlDataオブジェクト、
 2211…Attributeオブジェクトの属性値に該当するDataNodeの例、
 2301…エレメント「特許」に対応するElementNodeオブジェクトの例、
 2302…エレメント「明細書」に対応するElementNodeオブジェクトの例、
 2303…エレメント「図面」に対応するElementNodeオブジェクトの例、
 2304…文字列を内容データとするElementNodeオブジェクトの例、
 2305…DataNodeオブジェクトの例、
 2306…エレメント「図」に対応するElementNodeオブジェクトの例、
 2307…Attributeオブジェクトの例、
 2308…EntityNodeオブジェクトの例、
 2309…NonSgmlDataオブジェクトの例、
 2401…Transfer ElementオブジェクトのTargetOIDプロパティの仕様、
 2501…DocVersionオブジェクトの例、
 2502…HTML形式のデータを管理するRenditionオブジェクトの例、
 2503…SGML形式のデータを管理するRenditionオブジェクトの例、
 2504…JPEG形式のデータを管理するRenditionオブジェクトの例、
 2505…SGML宣言にアクセスするためのTransfer Elementオブジェクトの例、
 2506…論理構造の最上位ノードアクセスするためのTransfer Elementオブジェクトの例、

2507…DTDにアクセスするためのTransfer Elementオブジェクトの例、
 2508…SGML宣言オブジェクトの例、
 2509…DTDオブジェクトの例、
 2510…可視化情報オブジェクトの例、
 2511…ElementNodeオブジェクトの例、
 2512…Transfer Elementオブジェクトとそれが指し示すElementNodeとを示す。
 2513…2512を詳細に示したもの、
 2514…2506のTransfer Elementの内容を詳細に示したもの、
 2515…2511のElementNodeの内容を詳細に示したもの、
 2601…Transfer Elementにアクセスするステップ、
 2602…Transfer ElementのGetReferenceメソッドを呼ぶステップ、
 2603…Transfer Elementから論理構造の最上位のノードのオブジェクトIDを得るステップ、
 2604…文書空間オブジェクトのConnectObjectメソッドを呼ぶステップ、
 2605…文書空間オブジェクトから、該当するElementNodeを得るステップ、
 2701…DocVersionを取得するステップ、
 2702…DocVersionのRenditions Presentプロパティを取得するステップ、
 2703…望むRenditionを表す文字列の、Renditions Presentにおける位置を取得するステップ、
 2704…DocVersionのGetRenditionメソッドを呼ぶことにより、該当するRenditionを取得するステップ、
 2705…RenditionのTransfer Elements Presentプロパティを取得するステップ、
 2706…望むTransfer Elementを表す文字列の、Transfer Elements Presentにおける位置を取得するステップ、
 2707…RenditionのGetTransferElementメソッドを呼ぶことにより、該当するTransfer Elementを取得するステップ、
 2708…Transfer ElementのGetReferenceメソッドを呼ぶことにより、各Transfer Elementが該当する構成要素のObjectIDを得るステップ、
 2709…DocSpaceのConnectObjectメソッドを呼ぶことにより、Transfer Elementの指し示すオブジェクトを得るステップ、
 2710…取得したオブジェクトの種類を判定するステップ、
 2711…各オブジェクトのContentプロパティを取得することで、内容を得るステップ、

2712…取得したい情報を選択するステップ、
 2713…ElementNodeのContentsプロパティの値(SgmlNodeのリスト)を取得するステップ、2714…SgmlNodeのリストの中から望むSgmlNodeを取得するステップ、
 2715…SgmlNodeの種類を判定するステップ、
 2716…DataNodeからTextDataプロパティを取得し、内容文字列を得るステップ、
 2717…ElementのAttributesプロパティの値(Attributeのリスト)を取得するステップ、
 2718…Attributeのリストの中から望むAttributeを取得するステップ、
 2719…取得したAttributeから望むプロパティを取得するステップ、
 2720…属性値の種類を判定するステップ、
 2721…EntityNodeのNonSgmlDataプロパティの値であるNonSgmlDataオブジェクトを得るステップ、
 2722…NonSgmlDataオブジェクトのBinaryDataプロパティを取得し、バイナリデータを得るステップ、
 2801…PIクラスのContentプロパティの仕様、
 2901…DocNodeクラスのDTDプロパティの仕様、
 2902…DocNodeクラスのSGML宣言プロパティの仕様、
 2903…DocNodeクラスのContentsプロパティの仕様、
 3001…DocVersionオブジェクトの例、
 3002…HTML形式のデータを管理するRenditionオブジェクトの例、
 3003…SGML形式のデータを管理するRenditionオブジェクトの例、
 3004…JPEG形式のデータを管理するRenditionオブジェクトの例、
 3005…論理構造の最上位ノードアクセスするためのTransfer Elementオブジェクトの例、
 3006…DocNodeオブジェクトの例、
 3007…Transfer Elementオブジェクトとそれが指し示すDocNodeとを示す。
 3008…3007を詳細に示したもの、
 3009…3005のTransfer Elementの内容を詳細に示したもの、
 3010…3005のDocNodeのOIIDプロパティの内容を詳細に示したもの、
 3011…DTDオブジェクトの例、
 3012…可視化情報オブジェクトの例、
 3013…SGML宣言オブジェクトの例、
 3014…PIオブジェクトの例、
 3015…ElementNodeオブジェクトの例。

【図1】



DNAの文書モデルの概要

【図2】

図2

	プロパティ名	データ型	Cardinality	説明
201	Children	Object (Containment-Relationship)	List	格納対象となるオブジェクトをこのContainerに結び付けるContainmentRelationshipのリスト
202	Parent	Object (Containment-Relationship)	Scalar	このContainerと、このContainerを格納するContainerとを結びつけるContainmentRelationship

Container(101)の主なプロパティ

【図3】

図3

	プロパティ名	データ型	Cardinality	説明
301	Head	Object (Containable)	Scalar	格納対象オブジェクト
302	Tail	Object (Container)	Scalar	格納対象オブジェクトを格納するContainer

ContainmentRelationship(102)の主なプロパティ

【図4】

図4

プロパティ名	データ型	Cardinality	説明
401 Parnet	オブジェクト (Containable)	Scolor	このDocVersionと、このDocVersionを格納するContainerとを結びつけるContainmentRelationship
402 Renditions Presetnt	文字列	List	このDocVersionにどのような表現形式が存在するかを示す文字列のリスト。個々の文字列はそれぞれRenditionに対応する。

Document Version(103)の主なプロパティ

【図5】

図5

プロパティ名	データ型	Cardinality	説明
501 Rendition Type	文字列	Scolor	このRenditionが管理する表現形式を示す文字列。
502 Transfer Elements Presetnt	文字列	List	このRenditionにどのような構成要素が存在するかを示す文字列のリスト。個々の文字列はそれぞれTransfer Elementに対応する。

Rendition(104)の主なプロパティ

【図6】

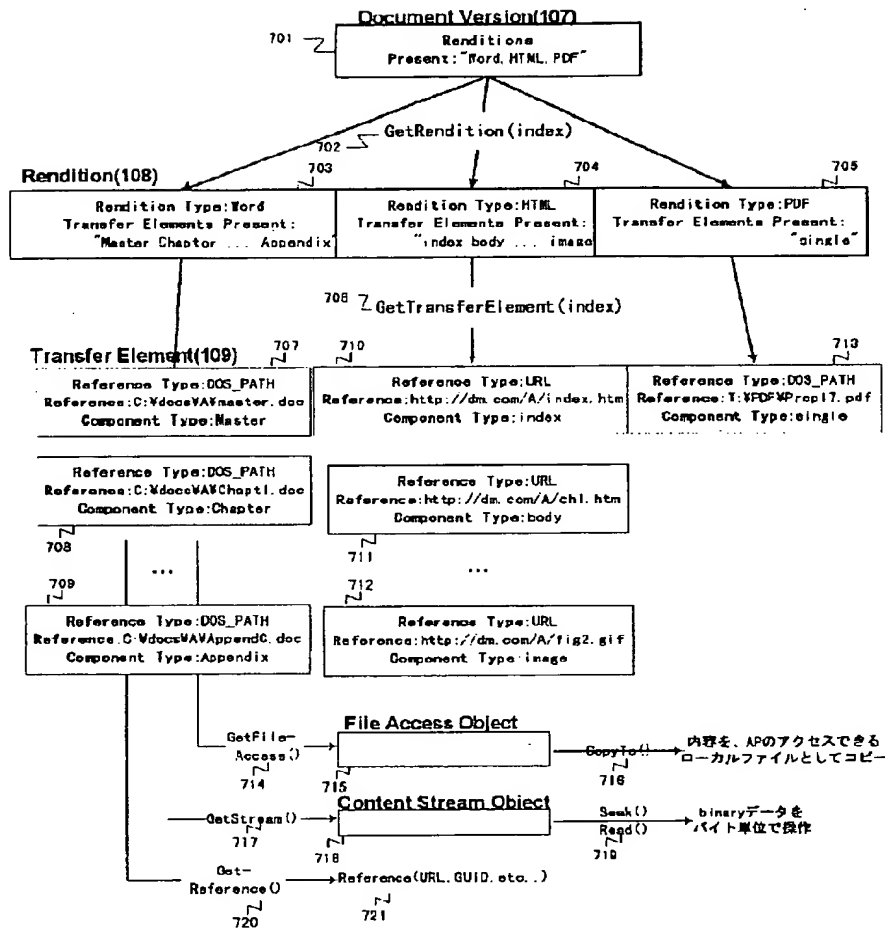
図6

プロパティ名	データ型	Cardinality	説明
601 Reference Type	整数	Scolor	Referenceの意味や文法を規定する整数。
602 Reference	文字列	List	内容データを格納する際にそのデータが存在していた位置へのリファレンス。
603 Component Type	文字列	Scolor	このTransfer Elementが管理する内容データの(あるRenditionの管理するデータ全体における)位置付けを示す文字列。

Transfer Element(105)の主なプロパティ

【図7】

図7



文書の内容管理の例

【図14】

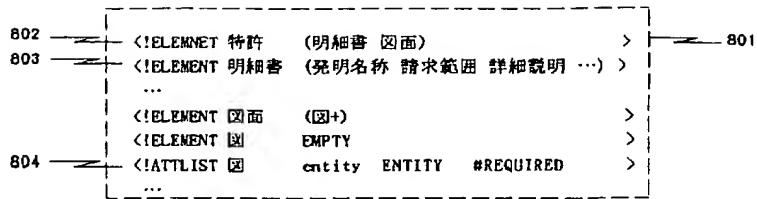
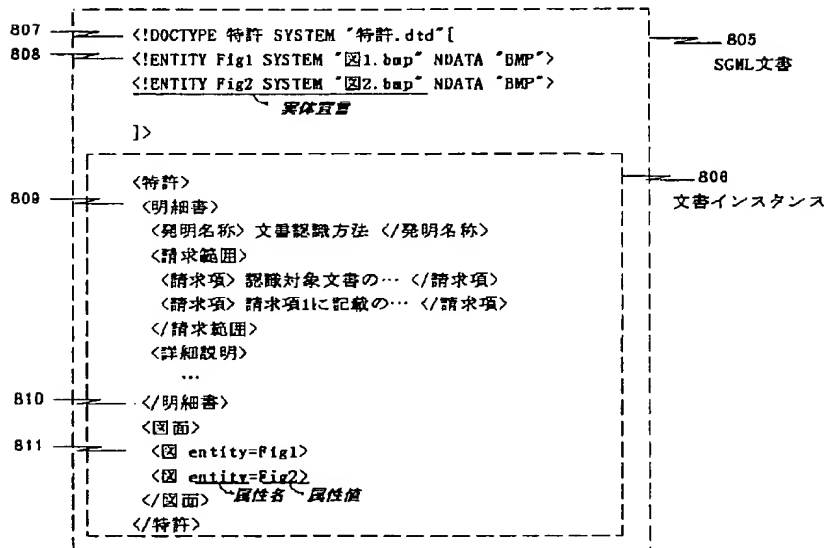
図14

	プロパティ名	データ型	Cardinality	説明
1401	Notation	文字列	Scalar	データの記法を示す文字列。
1402	BinaryData	バイナリ	Scalar	実データ。

NonXmlDataクラス(1303)の主なプロパティ

【図8】

図8

論理構造定義の例(特許.dtd)SGML文書の例

【図15】

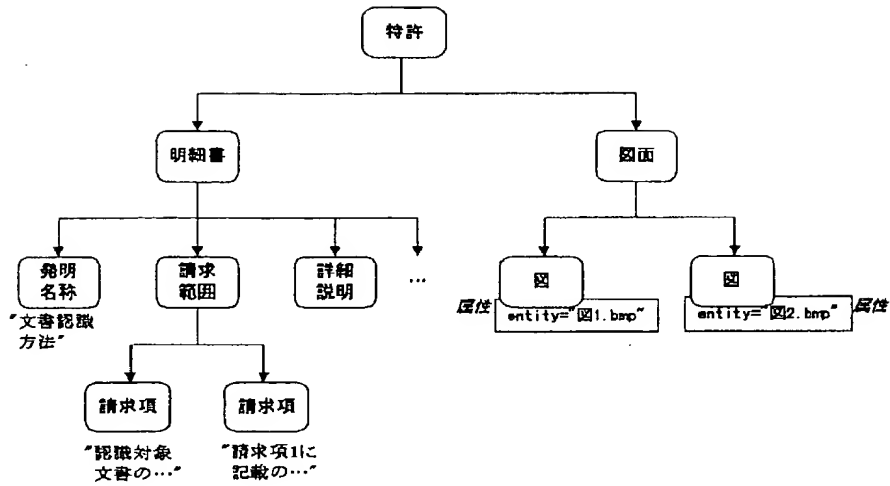
図15

	プロパティ名	データ型	Cardinality	説明
1501	Name	文字列	Scalar	このSGML宣言に付与する名称。
1502	Content	文字列	Scalar	テキスト形式のSGML宣言の実データ。

SGML宣言クラス(1304)の主なプロパティ

【図9】

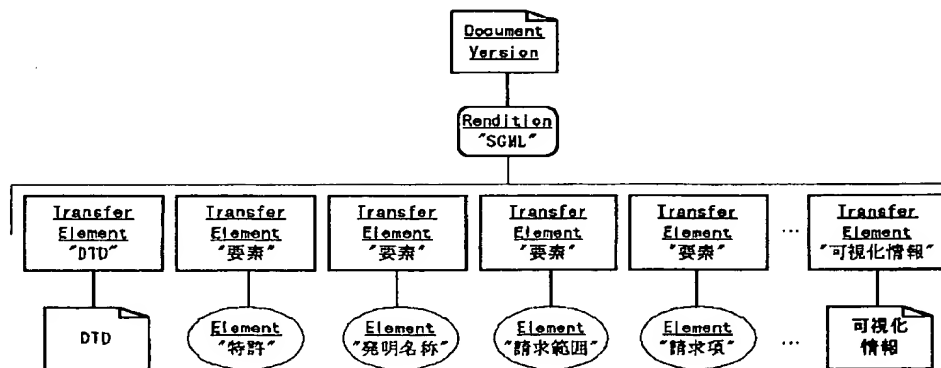
図9



SGML文書の論理的なツリー構造の例

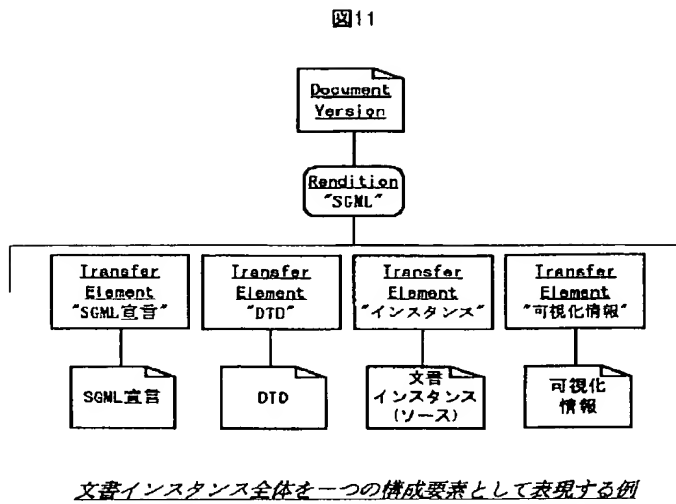
【図10】

図10

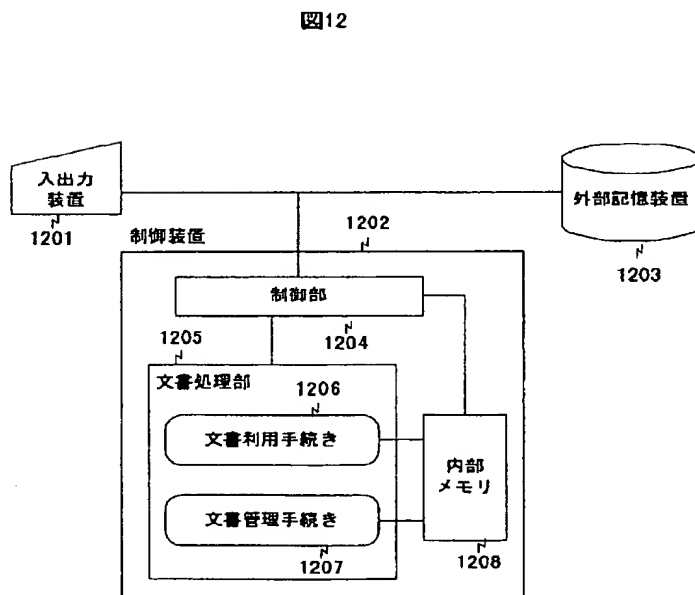


個々のエレメントにTransfer Elementを割りあてる例

【図11】

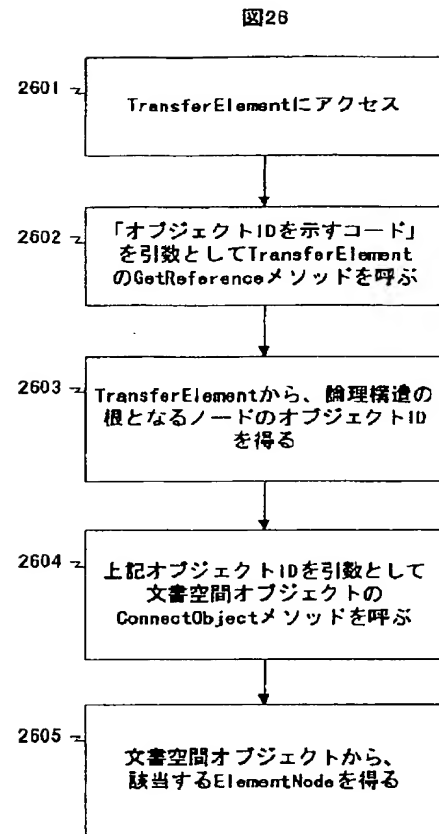


【図12】



装置構成例

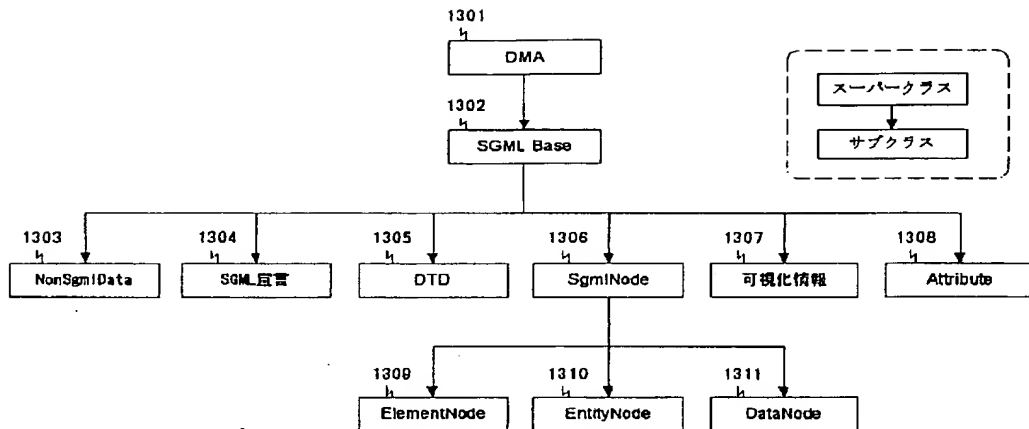
【図26】



アプリケーションが文書の論理構造の根となる
ElementNodeにアクセスするための処理

【図13】

図13



SGML文書を表現するためのオブジェクトのクラス

【図16】

図16

	プロパティ名	データ型	Cardinality	説明
1601	Name	文字列	Scalar	このDTDに付与する名称。
1602	Content	文字列	Scalar	テキスト形式のDTDの実データ。
1603	FormatInf	オブジェクト (可視化情報)	List	可視化情報オブジェクト。

DTDクラス(1305)の主なプロパティ

【図17】

図17

	プロパティ名	データ型	Cardinality	説明
1701	Name	文字列	Scalar	この可視化情報に付与する名称。
1702	Content	文字列	Scalar	テキスト形式の可視化情報の実データ。

可視化情報クラス(1307)の主なプロパティ

【図18】

図18

	プロパティ名	データ型	Cardinality	説明
1801	属性名	文字列	Scalar	この属性の名称。
1802	宣言値	文字列	Scalar	属性値の種類(型)を規定する情報。
1803	属性値	オブジェクト (DataNode or EntityNode)	Scalar	属性値の実データを保持するオブジェクト。

Attributeクラス(1308)の主なプロパティ

【図19】

図19

	プロパティ名	データ型	Cardinality	説明
1901	Name	文字列	Scalar	エレメントの名称。
1902	Contents	オブジェクト (SgmlNode)	List	エレメントの内容データを表現するSgmlNodeオブジェクトのリスト。
1903	Attributes	オブジェクト (Attribute)	List	エレメントの属性を表現するAttributeオブジェクトのリスト。

ElementNodeクラス(1309)の主なプロパティ

【図20】

図20

	プロパティ名	データ型	Cardinality	説明
2001	Name	文字列	Scalar	Entityの名称。
2002	NonSgmlData	オブジェクト (NonSgmlData)	Scalar	このエンティティ参照の参照対象となる非SGMLデータを保持するNonSgmlDataオブジェクト。

EntityNodeクラス(1310)の主なプロパティ

【図21】

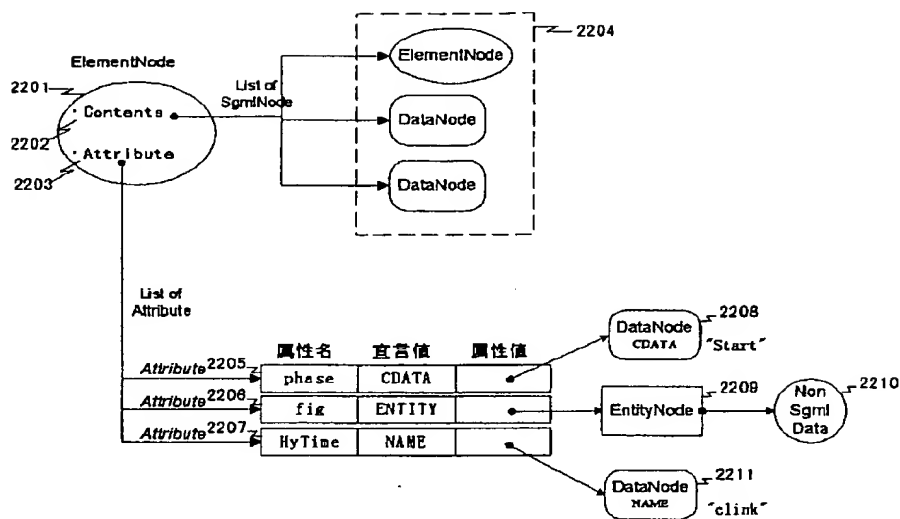
図21

	プロパティ名	データ型	Cardinality	説明
2101	Type	文字列	Scalar	文字列データの種類。
2102	TextData	文字列	Scalar	文字列データ。

DataNodeクラス(1311)の主なプロパティ

【図22】

図22

ElementNodeとAttributeの構成

【図24】

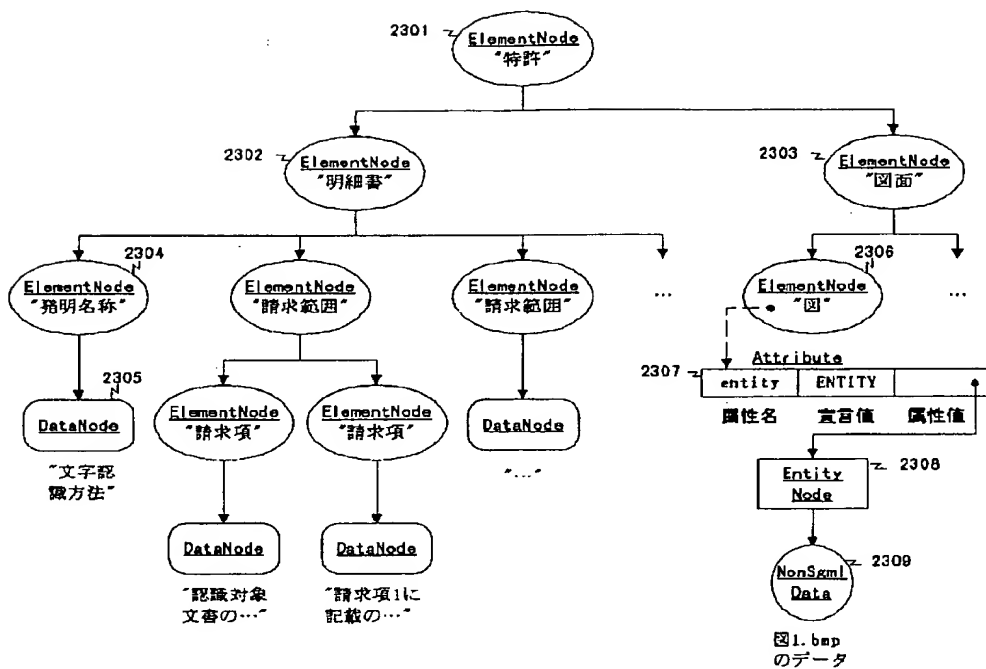
図24

プロパティ名	データ型	Cardinality	説明
Reference Type	整数	Scolor	Referenceの意味や文法を規定する整数。
Reference	文字列	List	内容データを格納する際にそのデータが存在していた位置へのリファレンス。
Component Type	文字列	Scolor	このTransfer Elementが管理する内容データの(あるRenditionの管理するデータ全体における)位置付けを示す文字列。
2401 TargetOID	OIID	Scolor	このTransfer Elementが管理する内容データの文書管理システム内部におけるオブジェクトID(OIID)。

Transfer Elementの実装(主なプロパティ)

【図23】

図23



SGML文書の論理構造のオブジェクトによる表現例

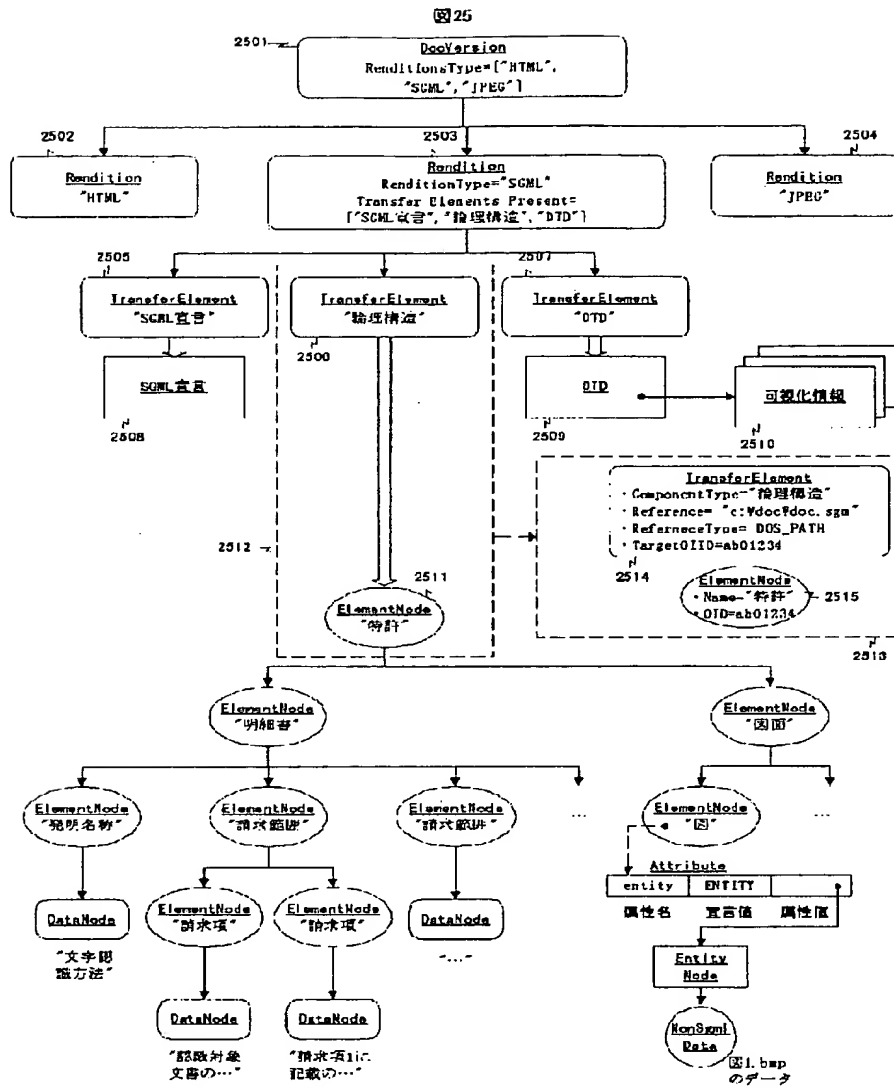
【図28】

図28

プロパティ名	データ型	Cardinality	説明
Content	文字列	Scalar	内容を表す文字列

PIクラスの主なプロパティ

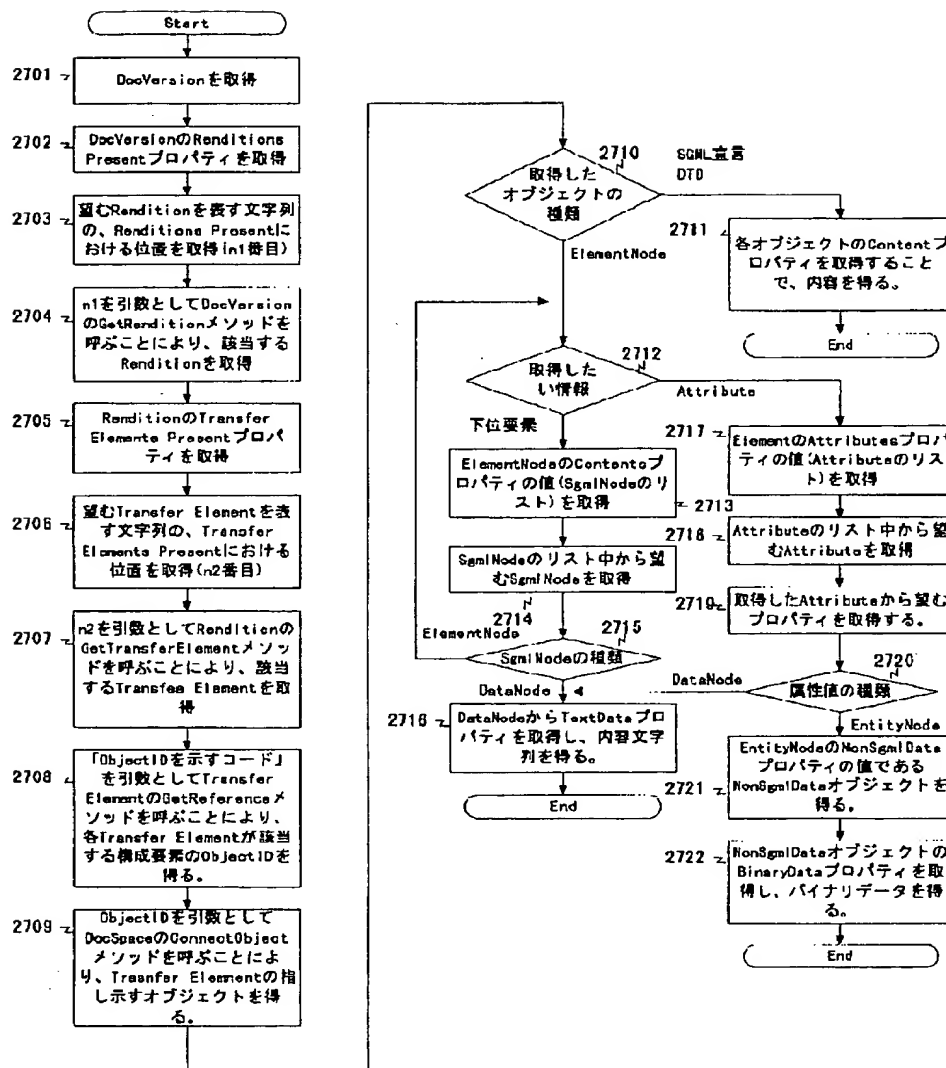
【図25】



拡張DAA文書モデルによるSGML文書の表現例

【図27】

図27



アプリケーションが文書の論理構造に
アクセスするための処理フロー

【図29】

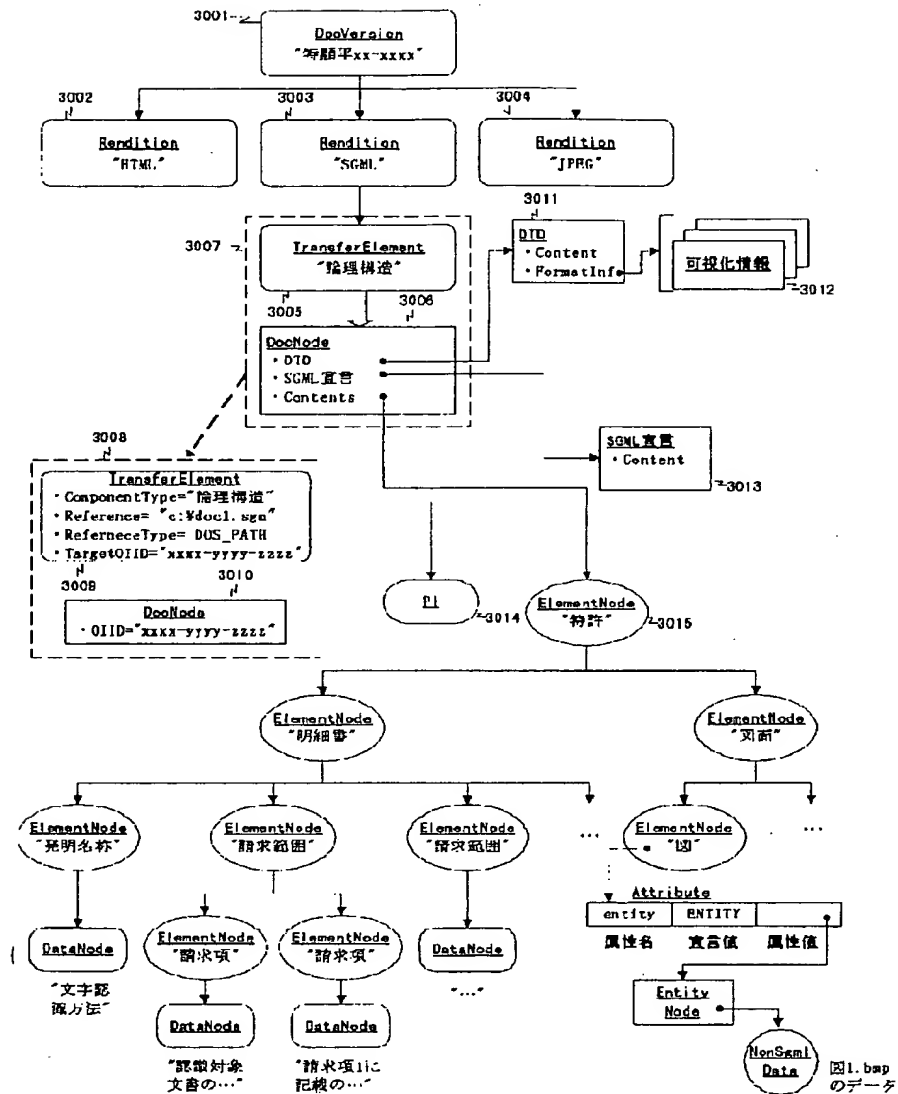
図29

	プロパティ名	データ型	Cardinality	説明
2901	DTD	OIID	Scalar	この文書のDTDを格納するDTDオブジェクトのOIID
2902	SGML宣言	OIID	Scalar	この文書のSGML宣言を格納するSGML宣言オブジェクトのOIID
2903	Contents	オブジェクト (SgmlNode)	List	文書の最上位エレメント(一つ)とPI(任意個)を保持するリスト

DocNodeクラスの主なプロパティ

【図30】

図30



単一のTransfer ElementによるSGML文書の表現例

フロントページの続き

(72)発明者 和歌山 哲

神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア開発本部内

(72)発明者 東 秋夫

東京都江東区新砂一丁目6番27号 株式会
社日立製作所公共情報事業部内